



WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

Bewertung von exotischen Optionen im CRR Modell

Bachelorarbeit

von
Stefanie Tiemann

11.08.2010

Betreuer: Privatdozent Dr. Volkert Paulsen
Institut für mathematische Statistik
Fachbereich Mathematik und Informatik
Westfälische Wilhelms-Universität Münster

Inhaltsverzeichnis

Einführung	1
1 Asiatische Optionen	3
Definition	3
Lemma 1	3
Bewertung asiatischer Optionen	4
R-Programm (asiatischeOptionen.R)	5
2 Lookback Optionen	8
Definition	8
Lemma 2	8
Bewertung von LookbackMaxOptionen	10
R-Programm (LookbackMaxOptionen1.R)	11
R-Programm (LookbackMaxOptionen2.R)	13
Bewertung von LookbackMinOptionen	15
R-Programm (LookbackMinOptionen1.R)	16
R-Programm (LookbackMinOptionen2.R)	18
3 Barriere Optionen	21
Definition (einseitige Barriere Optionen)	21
Definition (zweiseitige Barriere Optionen)	21
Lemma 3	22
Bewertung von knock-out Barriere Optionen ohne Rückvergütung	23
R-Programm (koBarriereOptionen.R)	25
Bewertung von einseitigen knock-in Barriere Optionen ohne Rückvergütung	27
R-Programme (up-and-in / down-and-in Barriere Option ohne Rückvergütung)	27
Lemma 4	28
Bewertung von zweiseitigen knock-in Barriere Optionen ohne Rückvergütung	29
R-Programm (zskiBarriereOptionen.R)	30
Bewertung von knock-in Barriere Optionen mit Rückvergütung	32
R-Programme (knock-in Barriere Optionen mit Rückvergütung)	32
Bewertung von einseitigen knock-out Barriere Optionen mit Rückvergütung	33
R-Programme (up-and-out / down-and-out Barriere Option mit Rückvergütung)	33
Bewertung von zweiseitigen knock-out Barriere Optionen mit Rückvergütung	33
R-Programm (zskoBarriereOptionen.R)	33
Fazit	36
Literaturverzeichnis	37

Einführung

In dieser Arbeit geht es um die Bewertung ausgewählter exotischer Optionen im CRR-Modell mithilfe der Markov-Eigenschaft. Die Arbeit umfasst drei Kapitel, in denen jeweils eine Optionsart betrachtet wird. Es werden asiatische Optionen, Lookback Optionen und Barriere Optionen behandelt.

Zu Beginn eines Kapitels wird die Optionsart definiert. Anschließend wird die Markov-Eigenschaft von zur Bewertung der jeweiligen Optionen benötigten stochastischen Prozessen nachgewiesen und damit Algorithmen zur Bewertung der Optionen hergeleitet. Diese Algorithmen sind in R implementiert, so dass mit den Programmen die Anfangspreise der betrachteten Optionen berechnet werden können.

Bei allen in dieser Arbeit behandelten Optionen handelt es sich um pfadabhängige exotische Optionen, d.h. um derivative Finanzgüter, deren Auszahlungen von einer Auszahlungsfunktion und der Kursentwicklung eines Underlyings während der Laufzeit der Option abhängen. Alle Optionen werden als Optionen europäischen Typs betrachtet, das bedeutet, dass bei allen Optionen eine endliche Laufzeit vorausgesetzt wird und eine Ausübung der Option nur am Ende ihrer Laufzeit möglich ist.

Exotische Optionen sind Finanzverträge, die in der Regel individuellen Kundenbedürfnissen entsprechen. Sie werden deshalb außerbörslich (OTC, over-the-counter) gehandelt.

Die Optionen werden in einem Cox-Ross-Rubinstein Modell (CRR Modell) bewertet, d.h. in einem diskreten N-Perioden-Finanzmarktmodell mit zwei Finanzgütern:

- risikofreie Anlage mit Preisprozess $(1 + \rho)^n$, $n = 0, \dots, N$, wobei $\rho > 0$ deterministische Zinsrate pro Periode ist
- risikobehaftetes Finanzgut mit Preisprozess
$$A_n = A_0 \cdot \prod_{i=1}^n Y_i, (Y_i)_{i=1, \dots, N} \text{ iid},$$
$$P(Y_i = u) = p = 1 - P(Y_i = d), 0 < d < u, A_0 > 0$$
- Informationsverlauf $\mathcal{F}_n = \sigma(A_0, A_1, \dots, A_n) = \sigma(A_0, Y_1, \dots, Y_n)$

Die Algorithmen zur Bewertung der Optionen werden mithilfe der Markov-Eigenschaft geeigneter Prozesse hergeleitet. Dabei wird die Markov-Eigenschaft in folgender Formulierung nachgewiesen:

Sei (Ω, \mathcal{F}, P) Wahrscheinlichkeitsraum. Ein zu einer Filtration $(\mathcal{F}_n)_{n \in \mathbb{N}}$ adaptierter Prozess $(X_{1,n}, \dots, X_{m,n})_{n \in \mathbb{N}}$ mit Werten in $(\mathbb{R}^m, \mathfrak{B}^m)$ erfüllt die Markov-Eigenschaft, falls für jede messbare Funktion $h : \mathbb{R}^m \rightarrow \mathbb{R}$, für die $E|h(X_{1,n+1}, \dots, X_{m,n+1})|$ existiert, gilt: $E[h(X_{1,n+1}, \dots, X_{m,n+1})|\mathcal{F}_n] = E[h(X_{1,n+1}, \dots, X_{m,n+1})|(X_{1,n}, \dots, X_{m,n})]$.

Für den Nachweis der Markov-Eigenschaft wird das folgende Lemma aus der Vorlesung Finanzmathematik I (WS 09/10) angewendet:

Lemma 0

Seien (M_1, \mathfrak{m}_1) , (M_2, \mathfrak{m}_2) messbare Räume, sei (Ω, \mathcal{F}, P) Wahrscheinlichkeitsraum, \mathcal{G} Unter- σ -Algebra von \mathcal{F} , seien $X_1 : \Omega \rightarrow M_1$, $X_2 : \Omega \rightarrow M_2$ und $f : (M_1 \times M_2, \mathfrak{m}_1 \otimes \mathfrak{m}_2) \rightarrow (\mathbb{R}, \mathfrak{B})$ messbare Abbildungen. Gelte ferner:

- (i) X_1 ist unabhängig von \mathcal{G}
- (ii) X_2 ist messbar bzgl. \mathcal{G}
- (iii) $Ef(X_1, X_2)$ ex.

Dann gilt: $E[f(X_1, X_2)|\mathcal{G}] = E[f(X_1, X_2)|X_2]$ P-f.s

1 Asiatische Optionen

Definition

Eine asiatische Option ist durch eine messbare Auszahlungsfunktion h bestimmt, die vom Mittelwert der Underlyingkurse während der Laufzeit der Option und gegebenenfalls dem Endkurs des Underlyings abhängt.

In der Praxis wird i.d.R. der arithmetische Mittelwert verwendet, d.h. $\frac{1}{N} \sum_{n=1}^N A_n$, wobei A_n der Underlyingkurs zum Zeitpunkt n ist und N die Laufzeit der Option.

Beispiele für Auszahlungsfunktionen sind:

- fixed strike Asian Call $(\frac{1}{N} \sum_{n=1}^N A_n - K)^+$
- fixed strike Asian Put $(K - \frac{1}{N} \sum_{n=1}^N A_n)^+$
- floating strike Asian Call $(A_N - \frac{1}{N} \sum_{n=1}^N A_n)^+$
- floating strike Asian Put $(\frac{1}{N} \sum_{n=1}^N A_n - A_N)^+$

Dabei ist eine feste Basis (fixed strike) eine bei Vertragsabschluss festgelegte Konstante K und eine variable Basis (floating strike) entspricht dem Mittelwert der Underlyingkurse. Ein Call bzw. ein Put ist bei fester Basis eine Absicherung des Mittelwerts der Underlyingkurse oberhalb bzw. unterhalb von K und bei variabler Basis eine Absicherung des Schlusskurses des Underlyings oberhalb bzw. unterhalb des Mittelwerts der Underlyingkurse.

Im Folgenden ist das Ziel die Bewertung asiatischer Optionen. Dazu wird folgendes Lemma benötigt:

Lemma 1

Sei $S_n := \sum_{k=0}^n A_k$ die laufende Summe der Underlyingkurse. Dann erfüllt der zweidimensionale Prozess $\{(A_n, S_n)\}_{n=0, \dots, N}$ die Markov-Eigenschaft.

Beweis:

Sei $n \in \{0, \dots, N-1\}$ beliebig.

Sei $h : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ eine messbare Funktion, für die $E|h(A_{n+1}, S_{n+1})|$ existiert.

Es gilt: $h(A_{n+1}, S_{n+1}) = h(A_{n+1}, S_n + A_{n+1}) = h(A_n \cdot Y_{n+1}, S_n + (A_n \cdot Y_{n+1}))$

Man definiere die Funktion $f : \mathbb{R} \times (\mathbb{R} \times \mathbb{R}) \rightarrow \mathbb{R}$,

$f(Y_{n+1}, (A_n, S_n)) := h(A_n \cdot Y_{n+1}, S_n + (A_n \cdot Y_{n+1})) = h(A_{n+1}, S_{n+1})$.

Dann gilt:

$$\begin{aligned} E[h(A_{n+1}, S_{n+1})|\mathcal{F}_n] &= E[f(Y_{n+1}, (A_n, S_n))|\mathcal{F}_n] \\ &\stackrel{\text{Lemma 0}}{=} E[f(Y_{n+1}, (A_n, S_n))|(A_n, S_n)] \\ &= E[h(A_{n+1}, S_{n+1})|(A_n, S_n)] \end{aligned}$$

Lemma 0 kann angewendet werden, da die Voraussetzungen des Lemmas erfüllt sind:

- $(\mathbb{R}, \mathfrak{B})$, $(\mathbb{R} \times \mathbb{R}, \mathfrak{B} \otimes \mathfrak{B})$ sind messbare Räume, (Ω, \mathcal{F}, P) ist Wahrscheinlichkeitsraum und \mathcal{F}_n Unter- σ -Algebra von \mathcal{F} .
- $Y_{n+1} : \Omega \rightarrow \mathbb{R}$, $(A_n, S_n) : \Omega \rightarrow \mathbb{R} \times \mathbb{R}$ sind messbare Abbildungen.
- $f : \mathbb{R} \times (\mathbb{R} \times \mathbb{R}) \rightarrow \mathbb{R}$, $f(Y_{n+1}, (A_n, S_n)) = h(A_{n+1}, S_{n+1})$ ist messbar.
- Y_{n+1} ist unabhängig von \mathcal{F}_n .
- (A_n, S_n) ist messbar bzgl. \mathcal{F}_n .
- $E f(Y_{n+1}, (A_n, S_n)) = E h(A_{n+1}, S_{n+1})$ existiert.

Somit erfüllt der zweidimensionale Prozess $\{(A_n, S_n)\}_{n=0, \dots, N}$ die Markov-Eigenschaft. \square

Bewertung asiatischer Optionen

Im Folgenden wird ein Algorithmus zur Bewertung asiatischer Optionen hergeleitet.

Sei $v_n(x, y)$ der Wert der Option in n , wenn $A_n = x$, $S_n = y$ gilt.

Sei P^* äquivalentes Martingalmaß mit $P^*(Y_i = u) = p^* = 1 - P^*(Y_i = d)$.

Eine asiatische Option entspricht dem Claim $C = \{C(n)\}_{n=1, \dots, N}$ mit $C(n) = 0$ für $n = 1, \dots, N-1$ und $C(N) = h(A_N, S_N)$.

Es gilt:

$$\begin{aligned}
v_0(A_0, S_0) &= E^*[(1 + \rho)^{-N} \cdot C(N)|(A_0, S_0)] \\
&= (1 + \rho)^{-N} \cdot E^*[C(N)|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[C(N)|\mathcal{F}_1]|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[h(A_N, S_N)|\mathcal{F}_1]|\mathcal{F}_0] \\
&\stackrel{\text{Lemma 1}}{=} (1 + \rho)^{-N} \cdot E^*[E^*[h(A_N, S_N)|(A_1, S_1)]|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[C(N)|(A_1, S_1)]|\mathcal{F}_0] \\
&= (1 + \rho)^{-1} \cdot E^*[E^*[(1 + \rho)^{-(N-1)} \cdot C(N)|(A_1, S_1)]|(A_0, S_0)] \\
&= (1 + \rho)^{-1} \cdot E^*[v_1(A_1, S_1)|(A_0, S_0)] \\
&= (1 + \rho)^{-1} \cdot (p^* \cdot v_1(u \cdot A_0, S_0 + u \cdot A_0) + (1 - p^*) \cdot v_1(d \cdot A_0, S_0 + d \cdot A_0))
\end{aligned}$$

Somit erhält man die Rekursionsformel:

$$v_n(A_n, S_n) = (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, S_n + u \cdot A_n) + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, S_n + d \cdot A_n))$$

für $n = N - 1, \dots, 0$.

Der Wert einer asiatischen Option in N entspricht der Auszahlung der Option in N , d.h. $v_N(A_N, S_N) = h(A_N, S_N)$.

Insgesamt ergibt sich der folgende Algorithmus zur Berechnung des Anfangspreises einer asiatischen Option:

Initialisierung: Für jedes mögliche Paar (A_N, S_N)

$$\text{berechne } v_N(A_N, S_N) = h(A_N, S_N)$$

Rückwärtsinduktion: Für $n = N - 1, \dots, 0$

Für jedes mögliche Paar (A_n, S_n)

$$\begin{aligned}
\text{berechne } v_n(A_n, S_n) &= (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, S_n + u \cdot A_n) \\
&\quad + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, S_n + d \cdot A_n))
\end{aligned}$$

Return: $v_0(A_0, S_0)$

$v_0(A_0, S_0)$ ist der Anfangspreis der Option.

R-Programm (asiatischeOptionen.R)

Mit dem folgenden Programm kann der Anfangspreis einer asiatischen Option berechnet werden. Dabei müssen die Modell-Parameter A_0 , u , d , ρ , N und die Auszahlungsfunktion h an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 20$.

```
# Definition des CRR-Modells
A0<-1          # Anfangskurs des Underlyings
u<-1.5         # up-Parameter des CRR-Modells
d<-0.6         # down-Parameter des CRR-Modells
rho<-0.05      # Zinsrate pro Periode
N<-20         # Anzahl Perioden
p<-(1+rho-d)/(u-d) # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

# Definition der Auszahlungsfunktion
# bspw. fixed strike Asian Call
K<-1          # Basis für Call
h<-function(A, D) {if(D<K){return(0)}
               else {return(D-K)}}

# A Underlyingkurs in N, D Mittelwert der Underlyingkurse

if (0<=p && p<=1) {
  e <- numeric()
  Aus <- array(e, dim=c(2^(N+1)-1,4))
  # Spalten: Periodenanzahl, Underlyingkurs, Summe Underlyingkurse, Optionswert
  # Speicherung in Aus:
  # Die Werte von Periode n stehen in den Zeilen 2^n bis 2^(n+1) - 1.
  # Steht An in Zeile 2^n+i, so steht dAn in Zeile 2^(n+1)+2i+1
  #                               und uAn in Zeile 2^(n+1)+2i.

  # Speichern des Startwerts
  Aus[1,1] <- 0
  Aus[1,2] <- A0
  Aus[1,3] <- A0

  # Ermittlung der möglichen Underlyingkurse und Summen
  Zeile <- 1          # Zeile im Array Aus
  for (n in 1:N) {for (i in (2^(n-1)):(2^n-1))
                  {Zeile <- Zeile + 1
                   Aus[Zeile,1] <- n
                   Aus[Zeile,2] <- u * Aus[i,2]
                   Aus[Zeile,3] <- Aus[i,3] + Aus[Zeile,2]

                   Zeile <- Zeile + 1
                   Aus[Zeile,1] <- n
```



```

        Aus[Zeile,2] <- d * Aus[i,2]
        Aus[Zeile,3] <- Aus[i,3] + Aus[Zeile,2] }}

# Initialisierung:
for (i in (2^(N)):(2^(N+1)-1)) {Aus[i,4] <- h(Aus[i,2], Aus[i,3]/N)}

# Rückwärtsinduktion:
for (n in seq(N-1,0,-1))
  {for (i in (2^(n)):(2^(n+1)-1))
    {Zeilevu <- 2^(n+1) + 2 * ( i - 2^n )
      Zeilevd <- Zeilevu + 1
      Aus[i,4] <- (1+rho)^(-1)*(p*Aus[Zeilevu,4]+(1-p)*Aus[Zeilevd,4]) }}

# Return:
cat ("Anfangspreis", Aus[1,4])  }

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}

```

2 Lookback Optionen

Definition

Eine Lookback Option ist durch eine messbare Auszahlungsfunktion h bestimmt, die vom für den Käufer der Option besten Underlyingkurs während der Laufzeit der Option und gegebenenfalls dem Endkurs des Underlyings abhängt. Dabei kann der beste Underlyingkurs aus Sicht des Käufers der Option abhängig von der Auszahlungsfunktion der minimale oder der maximale Underlyingkurs während der Laufzeit der Option sein. Lookback Optionen, deren Auszahlungen vom maximalen Underlyingkurs während der Laufzeit der Option abhängen, werden im Folgenden LookbackMaxOptionen genannt und Lookback Optionen, deren Auszahlungen vom minimalen Underlyingkurs während der Laufzeit der Option abhängen, werden im Folgenden LookbackMinOptionen genannt.

Beispiele für Auszahlungsfunktionen sind:

- fixed strike lookback Call $(\max_{n \in \{0, \dots, N\}} A_n - K)^+$
- fixed strike lookback Put $(K - \min_{n \in \{0, \dots, N\}} A_n)^+$
- floating strike lookback Call $(A_N - \min_{n \in \{0, \dots, N\}} A_n)^+$
- floating strike lookback Put $(\max_{n \in \{0, \dots, N\}} A_n - A_N)^+$

Im Folgenden ist das Ziel die Bewertung von Lookback Optionen. Dazu wird folgendes Lemma benötigt:

Lemma 2

Sei $M_n := \max_{k \in \{0, \dots, n\}} A_k$ das laufende Maximum der Underlyingkurse und

$m_n := \min_{k \in \{0, \dots, n\}} A_k$ das laufende Minimum der Underlyingkurse.

Dann erfüllen die zweidimensionalen Prozesse $\{(A_n, M_n)\}_{n=0, \dots, N}$ und $\{(A_n, m_n)\}_{n=0, \dots, N}$ die Markov-Eigenschaft.

Beweis:

Sei $n \in \{0, \dots, N-1\}$ beliebig.

Sei $h : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ eine messbare Funktion, für die $E|h(A_{n+1}, M_{n+1})|$ bzw.

$E|h(A_{n+1}, m_{n+1})|$ existiert.

Es gilt:

$h(A_{n+1}, M_{n+1}) = h(A_{n+1}, M_n \vee A_{n+1}) = h(A_n \cdot Y_{n+1}, M_n \vee (A_n \cdot Y_{n+1}))$ bzw.

$h(A_{n+1}, m_{n+1}) = h(A_{n+1}, m_n \wedge A_{n+1}) = h(A_n \cdot Y_{n+1}, m_n \wedge (A_n \cdot Y_{n+1}))$.

Man definiere die Funktion $f : \mathbb{R} \times (\mathbb{R} \times \mathbb{R}) \rightarrow \mathbb{R}$,

$f(Y_{n+1}, (A_n, M_n)) := h(A_n \cdot Y_{n+1}, M_n \vee (A_n \cdot Y_{n+1})) = h(A_{n+1}, M_{n+1})$ bzw.

$f(Y_{n+1}, (A_n, m_n)) := h(A_n \cdot Y_{n+1}, m_n \wedge (A_n \cdot Y_{n+1})) = h(A_{n+1}, m_{n+1})$.

Dann gilt:

$$\begin{aligned} E[h(A_{n+1}, M_{n+1})|\mathcal{F}_n] &= E[f(Y_{n+1}, (A_n, M_n))|\mathcal{F}_n] \\ &\stackrel{\text{Lemma 0}}{=} E[f(Y_{n+1}, (A_n, M_n))|(A_n, M_n)] \\ &= E[h(A_{n+1}, M_{n+1})|(A_n, M_n)] \end{aligned}$$

bzw.

$$\begin{aligned} E[h(A_{n+1}, m_{n+1})|\mathcal{F}_n] &= E[f(Y_{n+1}, (A_n, m_n))|\mathcal{F}_n] \\ &\stackrel{\text{Lemma 0}}{=} E[f(Y_{n+1}, (A_n, m_n))|(A_n, m_n)] \\ &= E[h(A_{n+1}, m_{n+1})|(A_n, m_n)] \end{aligned}$$

Lemma 0 kann angewendet werden, da die Voraussetzungen des Lemmas erfüllt sind:

- $(\mathbb{R}, \mathfrak{B})$ und $(\mathbb{R} \times \mathbb{R}, \mathfrak{B} \otimes \mathfrak{B})$ sind messbare Räume, (Ω, \mathcal{F}, P) ist Wahrscheinlichkeitsraum und \mathcal{F}_n Unter- σ -Algebra von \mathcal{F} .
- $Y_{n+1} : \Omega \rightarrow \mathbb{R}$ und $(A_n, M_n) : \Omega \rightarrow \mathbb{R} \times \mathbb{R}$ bzw. $(A_n, m_n) : \Omega \rightarrow \mathbb{R} \times \mathbb{R}$ sind messbare Abbildungen.
- $f : \mathbb{R} \times (\mathbb{R} \times \mathbb{R}) \rightarrow \mathbb{R}$, $f(Y_{n+1}, (A_n, M_n)) = h(A_{n+1}, M_{n+1})$ bzw. $f(Y_{n+1}, (A_n, m_n)) = h(A_{n+1}, m_{n+1})$ ist messbar.
- Y_{n+1} ist unabhängig von \mathcal{F}_n .
- (A_n, M_n) bzw. (A_n, m_n) ist messbar bzgl. \mathcal{F}_n .

- $Ef(Y_{n+1}, (A_n, M_n)) = Eh(A_{n+1}, M_{n+1})$ bzw. $Ef(Y_{n+1}, (A_n, m_n)) = Eh(A_{n+1}, m_{n+1})$ existiert.

Somit erfüllen die zweidimensionalen Prozesse $\{(A_n, M_n)\}_{n=0, \dots, N}$ und $\{(A_n, m_n)\}_{n=0, \dots, N}$ die Markov-Eigenschaft. \square

Bewertung von LookbackMaxOptionen

Im Folgenden wird ein Algorithmus zur Bewertung von LookbackMaxOptionen hergeleitet.

Sei $v_n(x, y)$ der Wert der Option in n , wenn $A_n = x$, $M_n = y$ gilt.

Sei P^* äquivalentes Martingalmaß mit $P^*(Y_i = u) = p^* = 1 - P^*(Y_i = d)$.

Eine LookbackMaxOption entspricht dem Claim $C = \{C(n)\}_{n=1, \dots, N}$ mit $C(n) = 0$ für $n = 1, \dots, N - 1$ und $C(N) = h(A_N, M_N)$.

Es gilt:

$$\begin{aligned}
v_0(A_0, M_0) &= E^*[(1 + \rho)^{-N} \cdot C(N)|(A_0, M_0)] \\
&= (1 + \rho)^{-N} \cdot E^*[C(N)|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[C(N)|\mathcal{F}_1]|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[h(A_N, M_N)|\mathcal{F}_1]|\mathcal{F}_0] \\
&\stackrel{\text{Lemma 2}}{=} (1 + \rho)^{-N} \cdot E^*[E^*[h(A_N, M_N)|(A_1, M_1)]|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[C(N)|(A_1, M_1)]|\mathcal{F}_0] \\
&= (1 + \rho)^{-1} \cdot E^*[E^*[(1 + \rho)^{-(N-1)} \cdot C(N)|(A_1, M_1)]|(A_0, M_0)] \\
&= (1 + \rho)^{-1} \cdot E^*[v_1(A_1, M_1)|(A_0, M_0)] \\
&= (1 + \rho)^{-1} \cdot (p^* \cdot v_1(u \cdot A_0, M_0 \vee u \cdot A_0) + (1 - p^*) \cdot v_1(d \cdot A_0, M_0 \vee d \cdot A_0)) \\
&= (1 + \rho)^{-1} \cdot (p^* \cdot v_1(u \cdot A_0, M_0 \vee u \cdot A_0) + (1 - p^*) \cdot v_1(d \cdot A_0, M_0))
\end{aligned}$$

Somit erhält man die Rekursionsformel:

$$v_n(A_n, M_n) = (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, M_n \vee u \cdot A_n) + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, M_n)) \text{ für } n = N - 1, \dots, 0.$$

Der Wert einer Lookback Option in N entspricht der Auszahlung der Option in N , d.h. $v_N(A_N, M_N) = h(A_N, M_N)$.

Insgesamt ergibt sich der folgende Algorithmus zur Berechnung des Anfangspreises einer LookbackMaxOption:

Initialisierung: Für jedes mögliche Paar (A_N, M_N)

$$\text{berechne } v_N(A_N, M_N) = h(A_N, M_N)$$

Rückwärtsinduktion: Für $n = N - 1, \dots, 0$

Für jedes mögliche Paar (A_n, M_n)

$$\begin{aligned} \text{berechne } v_n(A_n, M_n) = & (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, M_n \vee u \cdot A_n) \\ & + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, M_n)) \end{aligned}$$

Return: $v_0(A_0, M_0)$

$v_0(A_0, M_0)$ ist der Anfangspreis der Option.

R-Programme (LookbackMaxOptionen1.R, LookbackMaxOptionen2.R)

Mit dem folgenden Programm (LookbackMaxOptionen1.R) kann der Anfangspreis einer LookbackMaxOption berechnet werden. Dabei müssen die Modell-Parameter A_0 , u , d , ρ , N und die Auszahlungsfunktion h an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 100$.

```
# Definition des CRR-Modells
A0<-1          # Anfangskurs des Underlyings
u<-1.5         # up-Parameter des CRR-Modells
d<-0.6         # down-Parameter des CRR-Modells
rho<-0.05     # Zinsrate pro Periode
N<-20         # Anzahl Perioden
p<-(1+rho-d)/(u-d) # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

# Definition der Auszahlungsfunktion
# bspw. fixed strike lookback Call
K<-1          # Basis für Call
h<-function(A,M) {if(M<K){return(0)}
               else {return(M-K)}}

# A Underlyingkurs in N, M Maximum der Underlyingkurse

if (0<=p && p<=1) {
e <- numeric()
Aus <- array(e, dim=c(10000000,5))
# Spalten: Periodenanzahl, Anzahl Ups im Underlyingkurs, Underlyingkurs,
```

```

#           Maximum Underlyingkurse, Optionswert

LowHigh <- array(e, dim=c(N,N+1,2))
# Zum Merken der Positionen der Daten der Perioden und Ups im array Aus
# Dimension 1: Periode, Dimension 2: Anzahl ups im Underlyingkurs + 1,
# Dimension 3 enthält die 2 Informationen: erste Zeile, letzte Zeile

Zeile <- 0           # Zeile im Array Aus

# Ermittlung der möglichen Underlyingkurse und Maxima
for (n in 1:N)
  {for (j in 0:n)
    {LowHigh[n,j+1,1] <- Zeile + 1           # Daten Periode n, j ups
      Zeile <- Zeile + 1
      Aus[Zeile,1] <- n                       # Periodenanzahl
      Aus[Zeile,2] <- j                       # Anzahl Ups im Underlyingkurs
      Kurs <- A0 * u ^ j * d ^ (n-j)         # Underlyingkurs
      Aus[Zeile,3] <- Kurs
      if (Kurs > A0) {Aus[Zeile,4] <- Kurs}   # Erstes mögliches Maximum
      else {Aus[Zeile,4] <- A0}              # Anfangskurs oder Endkurs

# Ermittlung weiterer möglicher Maxima
    {for (i1 in 0:j)
      {for (i2 in 0:(n-j))
        {ZwischenKurs <- A0 * u ^ i1 * d ^ i2
          if (ZwischenKurs > A0 && ZwischenKurs > Kurs)
            {Zeile <-Zeile+1
              Aus[Zeile,1] <- n
              Aus[Zeile,2] <- j
              Aus[Zeile,3] <- Kurs
              Aus[Zeile,4] <- ZwischenKurs}
            else {break} }}}
      LowHigh[n,j+1,2] <- Zeile }}

# Initialisierung:
for (i in LowHigh[N,1,1]:LowHigh[N,N+1,2]) {Aus[i,5] <- h(Aus[i,3], Aus[i,4])}

# Rückwärtsinduktion:
for (n in (seq(N-1,1,-1)))
  {for (i in LowHigh[n,1,1]:LowHigh[n,n+1,2])
    {uAn <- u*Aus[i,3]
      Mn <- max(Aus[i,4],uAn)

      for (j in LowHigh[n+1,Aus[i,2]+1,1]:LowHigh[n+1,Aus[i,2]+1,2])
        {if (Aus[j,4] == Aus[i,4]) {vd <- Aus[j,5]} }
    }
  }

```

```

#
#       for (j in LowHigh[n+1,Aus[i,2]+2,1]:LowHigh[n+1,Aus[i,2]+2,2])
#           {if (Aus[j,4] == Mn)   {vu <- Aus[j,5]} }   (*)
#
# Beim Vergleich if (Aus[j,4] == Mn) ergibt sich das Problem, dass beide Werte
# teilweise minimal voneinander abweichen.
# Deshalb ist (*) wie folgt ersetzt:
#
#       diff <- 1000
#       for (j in LowHigh[n+1,Aus[i,2]+2,1]:LowHigh[n+1,Aus[i,2]+2,2])
#           {if (abs(Aus[j,4] - Mn) < diff)
#               {vu <- Aus[j,5]
#                 diff <- abs(Aus[j,4] - Mn)} }
#
#       Aus[i,5] <- (1+rho)^(-1)*(p*vu+(1-p)*vd) }
#
# Berechnung des Anfangspreises der Option
v0 <- (1+rho)^(-1)*(p*Aus[2,5]+(1-p)*Aus[1,5])
#
# Return:
cat ("Anfangspreis", v0)}

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}

```

Mit dem folgenden Programm (LookbackMaxOptionen2.R) kann der Anfangspreis einer LookbackMaxOption für den Spezialfall $u = \frac{1}{d}$ berechnet werden. Dabei müssen die Modell-Parameter A_0 , u , ρ , N und die Auszahlungsfunktion h an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 450$.

```

# Definition des CRR-Modells
A0<-1           # Anfangskurs des Underlyings
u<-1.5          # up-Parameter des CRR-Modells
d<-1/u         # down-Parameter des CRR-Modells
rho<-0.05      # Zinsrate pro Periode
N<-50          # Anzahl Perioden
p<-(1+rho-d)/(u-d) # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

# Definition Auszahlungsfunktion
# bspw. fixed strike lookback Call
K<-1           # Basis für Call
h<-function(A,M) {if(M<K){return(0)}
                else {return(M-K)}}
# A Underlyingkurs in N, M Maximum der Underlyingkurse

```

```

if (0<=p && p<=1) {
e <- numeric()
Aus <- array(e, dim=c(1000000,6))
# Spalten: Periodenanzahl, Anzahl Ups im Underlyingkurs, Underlyingkurs,
#           Maximum Underlyingkurse, Optionswert, Anzahl Ups im Maximum

LowHigh <- array(e, dim=c(N,N+1,3))
# Zum Merken der Positionen der Daten der Perioden und Ups im array Aus
# Dimension 1: Periode, Dimension 2: Anzahl ups im Underlyingkurs + 1,
# Dimension 3 enthält die 3 Informationen:
# erste Zeile, letzte Zeile, Anzahl Ups im Maximum der ersten Zeile

Zeile <- 0           # Zeile im Array Aus

# Ermittlung der möglichen Underlyingkurse und Maxima
for (n in 1:N)
  {for (j in 0:n)
    {LowHigh[n,j+1,1] <- Zeile + 1      # Daten Periode n, j ups
      Zeile <- Zeile + 1
      Aus[Zeile,1] <- n                 # Periodenanzahl
      Aus[Zeile,2] <- j                 # Anzahl Ups im Underlyingkurs
      Kurs <- A0 * u ^ (2*j-n)         # Underlyingkurs
      Aus[Zeile,3] <- Kurs
      if (Kurs > A0) {Aus[Zeile,4] <- Kurs
                      Aus[Zeile,6] <- 2*j-n
                      LowHigh[n,j+1,3] <- 2*j-n} # Erstes mögliches Maximum
                      else {Aus[Zeile,4] <- A0           # Anfangskurs oder Endkurs
                              Aus[Zeile,6] <- 0
                              LowHigh[n,j+1,3] <- 0}

# Ermittlung weiterer möglicher Maxima
for (i1 in 0:j)
  {ZwischenKurs <- A0 * u ^ i1
    if (ZwischenKurs > A0 && ZwischenKurs > Kurs)
      {Zeile <-Zeile+1
        Aus[Zeile,1] <- n
        Aus[Zeile,2] <- j
        Aus[Zeile,3] <- Kurs
        Aus[Zeile,4] <- ZwischenKurs
        Aus[Zeile,6] <- i1  }}

    LowHigh[n,j+1,2] <- Zeile  }}

# Initialisierung:
for (i in LowHigh[N,1,1]:LowHigh[N,N+1,2]) {Aus[i,5] <- h(Aus[i,3],Aus[i,4])}

```



```

# Rückwärtsinduktion:
for (n in (seq(N-1,1,-1)))
  {for (i in LowHigh[n,1,1]:LowHigh[n,n+1,2])
    {AnzdUp <- Aus[i,6]
      if (Aus[i,6] > 2*Aus[i,2] - Aus[i,1]) {AnzuUp <- Aus[i,6]}
      else {AnzuUp <- Aus[i,6] + 1}

      # Ermitteln der Zeile von vd in Periode n+1
      vdZeile <- LowHigh[n+1,Aus[i,2]+1,1] + AnzdUp - LowHigh[n+1,Aus[i,2]+1,3]

      # Ermitteln von vd in Periode n+1
      vd <- Aus[vdZeile,5]

      # Ermitteln der Zeile von vu in Periode n+1
      vuZeile <- LowHigh[n+1,Aus[i,2]+2,1] + AnzuUp - LowHigh[n+1,Aus[i,2]+2,3]

      # Ermitteln von vu in Periode n+1
      vu <- Aus[vuZeile,5]

      Aus[i,5] <- (1+rho)^(-1)*(p*vu+(1-p)*vd) }}

# Berechnung des Anfangspreises der Option
v0 <- (1+rho)^(-1)*(p*Aus[2,5]+(1-p)*Aus[1,5])

# Return:
cat ("Anfangspreis", v0)}

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}

```

Bewertung von LookbackMinOptionen

Im Folgenden wird ein Algorithmus zur Bewertung von LookbackMinOptionen hergeleitet.

Sei $v_n(x, y)$ der Wert der Option in n , wenn $A_n = x$, $m_n = y$ gilt.

Sei P^* äquivalentes Martingalmaß mit $P^*(Y_i = u) = p^* = 1 - P^*(Y_i = d)$.

Eine LookbackMinOption entspricht dem Claim $C = \{C(n)\}_{n=1,\dots,N}$ mit $C(n) = 0$ für $n = 1, \dots, N - 1$ und $C(N) = h(A_N, m_N)$.

Es gilt:

$$\begin{aligned}
 v_0(A_0, m_0) &= E^*[(1 + \rho)^{-N} \cdot C(N)|(A_0, m_0)] \\
 &= \dots \text{ analog zu } C(N) = h(A_N, M_N) \dots \\
 &= (1 + \rho)^{-1} \cdot E^*[v_1(A_1, m_1)|(A_0, m_0)] \\
 &= (1 + \rho)^{-1} \cdot (p^* \cdot v_1(u \cdot A_0, m_0 \wedge u \cdot A_0) + (1 - p^*) \cdot v_1(d \cdot A_0, m_0 \wedge d \cdot A_0)) \\
 &= (1 + \rho)^{-1} \cdot (p^* \cdot v_1(u \cdot A_0, m_0) + (1 - p^*) \cdot v_1(d \cdot A_0, m_0 \wedge d \cdot A_0))
 \end{aligned}$$

Somit erhält man die Rekursionsformel:

$$v_n(A_n, m_n) = (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, m_n) + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, m_n \wedge d \cdot A_n)) \text{ für } n = N - 1, \dots, 0.$$

Der Wert einer Lookback Option in N entspricht der Auszahlung der Option in N, d.h. $v_N(A_N, m_N) = h(A_N, m_N)$.

Insgesamt ergibt sich der folgende Algorithmus zur Berechnung des Anfangspreises einer LookbackMinOption:

Initialisierung: Für jedes mögliche Paar (A_N, m_N)

$$\text{berechne } v_N(A_N, m_N) = h(A_N, m_N)$$

Rückwärtsinduktion: Für $n = N - 1, \dots, 0$

Für jedes mögliche Paar (A_n, m_n)

$$\begin{aligned}
 \text{berechne } v_n(A_n, m_n) &= (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, m_n) \\
 &\quad + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, m_n \wedge d \cdot A_n))
 \end{aligned}$$

Return: $v_0(A_0, m_0)$

$v_0(A_0, m_0)$ ist der Anfangspreis der Option.

R-Programme (LookbackMinOptionen1.R, LookbackMinOptionen2.R)

Mit dem folgenden Programm (LookbackMinOptionen1.R) kann der Anfangspreis einer LookbackMinOption berechnet werden. Dabei müssen die Modell-Parameter A_0 , u , d , ρ , N und die Auszahlungsfunktion h an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 100$.

```

# Definition des CRR-Modells
A0<-1          # Anfangskurs des Underlyings
u<-1.5         # up-Parameter des CRR-Modells
d<-0.6         # down-Parameter des CRR-Modells
rho<-0.05      # Zinsrate pro Periode
N<-20         # Anzahl Perioden
p<-(1+rho-d)/(u-d) # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

# Definition der Auszahlungsfunktion
# bspw. fixed strike lookback Put
K<-1          # Basis für Put
h<-function(A,m) {if(m>K){return(0)}
               else {return(K-m)}}
# A Underlyingkurs in N, m Minimum der Underlyingkurse

if (0<=p && p<=1) {
e <- numeric()
Aus <- array(e, dim=c(10000000,5))
# Spalten: Periodenanzahl, Anzahl Ups im Underlyingkurs, Underlyingkurs,
#         Minimum Underlyingkurse, Optionswert

LowHigh <- array(e, dim=c(N,N+1,2))
# Zum Merken der Positionen der Daten der Perioden und Ups im array Aus
# Dimension 1: Periode, Dimension 2: Anzahl ups im Underlyingkurs + 1,
# Dimension 3 enthält die 2 Informationen: erste Zeile, letzte Zeile

Zeile <- 0          # Zeile im Array Aus

# Ermittlung der möglichen Underlyingkurse und Minima
for (n in 1:N)
  {for (j in 0:n)
    {LowHigh[n,j+1,1] <- Zeile + 1          # Daten Periode n, j ups
      Zeile <- Zeile + 1
      Aus[Zeile,1] <- n                    # Periodenanzahl
      Aus[Zeile,2] <- j                    # Anzahl Ups im Underlyingkurs
      Kurs <- A0 * u ^ j * d ^ (n-j)      # Underlyingkurs
      Aus[Zeile,3] <- Kurs
      if (Kurs > A0) {Aus[Zeile,4] <- A0} # Erstes mögliches Minimum
        else {Aus[Zeile,4] <- Kurs}      # Anfangskurs oder Endkurs

# Ermittlung weiterer möglicher Minima
  {for (i1 in 0:j)
    {for (i2 in seq((n-j),0,-1))
      {ZwischenKurs <- A0 * u ^ i1 * d ^ i2
        if (ZwischenKurs < A0 && ZwischenKurs < Kurs)
          {Zeile <-Zeile+1

```

```

        Aus[Zeile,1] <- n
        Aus[Zeile,2] <- j
        Aus[Zeile,3] <- Kurs
        Aus[Zeile,4] <- ZwischenKurs}
    else { break } }}}}
LowHigh[n,j+1,2] <- Zeile  }

# Initialisierung:
for (i in LowHigh[N,1,1]:LowHigh[N,N+1,2]) {Aus[i,5] <- h(Aus[i,3],Aus[i,4])}

# Rückwärtsinduktion:
for (n in (seq(N-1,1,-1)))
  {for (i in LowHigh[n,1,1]:LowHigh[n,n+1,2])
    {dAn <- d*Aus[i,3]
    mn <- min(Aus[i,4],dAn)

    for (j in LowHigh[n+1,Aus[i,2]+2,1]:LowHigh[n+1,Aus[i,2]+2,2])
      {if (Aus[j,4] == Aus[i,4]) {vu <- Aus[j,5]} }

    diff <- 1000
    for (j in LowHigh[n+1,Aus[i,2]+1,1]:LowHigh[n+1,Aus[i,2]+1,2])
      {if (abs(Aus[j,4] - mn) < diff)
        {vd <- Aus[j,5]
        diff <- abs(Aus[j,4] - mn)} }

    Aus[i,5] <- (1+rho)^(-1)*(p*vu+(1-p)*vd) }}

# Berechnung des Anfangspreises der Option
v0 <- (1+rho)^(-1)*(p*Aus[2,5]+(1-p)*Aus[1,5])

# Return:
cat ("Anfangspreis", v0)}

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}
```

Mit dem folgenden Programm (LookbackMinOptionen2.R) kann der Anfangspreis einer LookbackMinOption für den Spezialfall $u = \frac{1}{d}$ berechnet werden. Dabei müssen die Modell-Parameter A_0 , u , ρ , N und die Auszahlungsfunktion h an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 450$.

```

# Definition des CRR-Modells
A0<-1          # Anfangskurs des Underlyings
u<-1.5         # up-Parameter des CRR-Modells
d<-1/u        # down-Parameter des CRR-Modells
rho<-0.05     # Zinsrate pro Periode
```

```

N<-20                # Anzahl Perioden
p<-(1+rho-d)/(u-d)   # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

# Definition Auszahlungsfunktion
# bspw. fixed strike lookback Put
K<-1                 # Basis für Put
h<-function(A,m) {if(m>K){return(0)}
                  else {return(K-m)}}
# A Underlyingkurs in N, m Minimum der Underlyingkurse

if (0<=p && p<=1) {
e <- numeric()
Aus <- array(e, dim=c(1000000,6))
# Spalten: Periodenanzahl, Anzahl Ups im Underlyingkurs, Underlyingkurs,
#          Minimum Underlyingkurse, Optionswert, Anzahl Downs im Minimum

LowHigh <- array(e, dim=c(N,N+1,3))
# Zum Merken der Positionen der Daten der Perioden und Ups im array Aus
# Dimension 1: Periode, Dimension 2: Anzahl ups im Underlyingkurs + 1,
# Dimension 3 enthält die 3 Informationen:
# erste Zeile, letzte Zeile, Anzahl Downs im Minimum der ersten Zeile

Zeile <- 0           # Zeile im Array Aus

# Ermittlung der möglichen Underlyingkurse und Minima
for (n in 1:N)
  {for (j in 0:n)
    {LowHigh[n,j+1,1] <- Zeile + 1      # Daten Periode n, j ups
      Zeile <- Zeile + 1
      Aus[Zeile,1] <- n                 # Periodenanzahl
      Aus[Zeile,2] <- j                 # Anzahl Ups im Underlyingkurs
      Kurs <- A0 * u ^ (2*j-n)         # Underlyingkurs
      Aus[Zeile,3] <- Kurs
      if (Kurs < A0) {Aus[Zeile,4] <- Kurs
                      Aus[Zeile,6] <- n-2*j
                      LowHigh[n,j+1,3] <- n-2*j} # Erstes mögliches Minimum
                    else {Aus[Zeile,4] <- A0      # Anfangskurs oder Endkurs
                          Aus[Zeile,6] <- 0
                          LowHigh[n,j+1,3] <- 0}

# Ermittlung weiterer möglicher Minima
for (i1 in 1:n-j)
  {ZwischenKurs <- A0 * u ^ (-i1)
    if (ZwischenKurs < A0 && ZwischenKurs < Kurs)
      {Zeile <-Zeile+1
        Aus[Zeile,1] <- n

```

```

        Aus[Zeile,2] <- j
        Aus[Zeile,3] <- Kurs
        Aus[Zeile,4] <- ZwischenKurs
        Aus[Zeile,6] <- i1  }}

    LowHigh[n,j+1,2] <- Zeile  }}

# Initialisierung:
for (i in LowHigh[N,1,1]:LowHigh[N,N+1,2]) {Aus[i,5] <- h(Aus[i,3],Aus[i,4])}

# Rückwärtsinduktion:
for (n in seq(N-1,1,-1))
  {for (i in LowHigh[n,1,1]:LowHigh[n,n+1,2])
    {AnzuDown <- Aus[i,6]
     if (Aus[i,6] > Aus[i,1] - 2*Aus[i,2]) {AnzdDown <- Aus[i,6]}
     else {AnzdDown <- Aus[i,6] + 1}

     # Ermitteln der Zeile von vd in Periode n+1
     vdZeile <- LowHigh[n+1,Aus[i,2]+1,1] + AnzdDown - LowHigh[n+1,Aus[i,2]+1,3]

     # Ermitteln von vd in Periode n+1
     vd <- Aus[vdZeile,5]

     # Ermitteln der Zeile von vu in Periode n+1
     vuZeile <- LowHigh[n+1,Aus[i,2]+2,1] + AnzuDown - LowHigh[n+1,Aus[i,2]+2,3]

     # Ermitteln von vu in Periode n+1
     vu <- Aus[vuZeile,5]

     Aus[i,5] <- (1+rho)^(-1)*(p*vu+(1-p)*vd)  }}

# Berechnung des Anfangspreises der Option
v0 <- (1+rho)^(-1)*(p*Aus[2,5]+(1-p)*Aus[1,5])

# Return:
cat ("Anfangspreis", v0)}

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}

```

3 Barriere Optionen

Man unterscheidet ein- und zweiseitige Barriere Optionen.

Definition (einseitige Barriere Optionen)

Einseitige Barriere Optionen sind durch eine Schranke s und eine messbare Auszahlungsfunktion h bestimmt.

Einseitige knock-out Barriere Optionen verfallen, wenn der Underlyingkurs während der Laufzeit der Option die Schranke s berührt oder überschreitet. Falls $A_0 < s$ nennt man die Option up-and-out Barriere Option und falls $A_0 > s$ down-and-out Barriere Option. Einseitige knock-in Barriere Optionen sind wertlos, solange der Underlyingkurs während der Laufzeit der Option die Schranke s weder berührt noch überschreitet. Falls $A_0 < s$ nennt man die Option up-and-in Barriere Option und falls $A_0 > s$ down-and-in Barriere Option.

Es gibt einseitige Barriere Optionen mit und ohne Rückvergütung (rebate). Eine Rückvergütung R ist eine konstante Auszahlung, die der Inhaber der Option im Fall des Verfalls einer knock-out Barriere Option bzw. bei Wertlosigkeit einer knock-in Barriere Option am Ende ihrer Laufzeit erhält. Die Auszahlung der Rückvergütung kann bei knock-out Barriere Optionen entweder bei Berühren der Schranke (payment at hit) oder am Ende der Laufzeit (payment at expiry) erfolgen, bei knock-in Barriere Optionen hingegen nur am Ende der Laufzeit. In dieser Arbeit wird bei einseitigen Barriere Optionen mit Rückvergütung der Fall payment at expiry vorausgesetzt.

Folgende Auszahlungen sind somit bei einseitigen Barriere Optionen am Ende der Laufzeit möglich:

	ohne Rückvergütung	zusätzliche Auszahlung bei Optionen mit Rückvergütung
up-and-out	$h(A_N) \mathbb{1}_{\{\max_{n \in \{0, \dots, N\}} A_n < s\}}$	$+R \mathbb{1}_{\{\max_{n \in \{0, \dots, N\}} A_n \geq s\}}$
down-and-out	$h(A_N) \mathbb{1}_{\{\min_{n \in \{0, \dots, N\}} A_n > s\}}$	$+R \mathbb{1}_{\{\min_{n \in \{0, \dots, N\}} A_n \leq s\}}$
up-and-in	$h(A_N) \mathbb{1}_{\{\max_{n \in \{0, \dots, N\}} A_n \geq s\}}$	$+R \mathbb{1}_{\{\max_{n \in \{0, \dots, N\}} A_n < s\}}$
down-and-in	$h(A_N) \mathbb{1}_{\{\min_{n \in \{0, \dots, N\}} A_n \leq s\}}$	$+R \mathbb{1}_{\{\min_{n \in \{0, \dots, N\}} A_n > s\}}$

Definition (zweiseitige Barriere Optionen)

Zweiseitige Barriere Optionen sind durch zwei Schranken l und r mit $l < A_0 < r$ und eine Auszahlungsfunktion h bestimmt.

Zweiseitige knock-out Barriere Optionen verfallen, wenn der Underlyingkurs während der Laufzeit der Option das durch l und r definierte offene Intervall verlässt.

Zweiseitige knock-in Barriere Optionen sind wertlos, solange der Underlyingkurs während der Laufzeit der Option das durch l und r definierte offene Intervall nicht verlässt.

Es gibt zweiseitige Barriere Optionen mit und ohne Rückvergütung. In dieser Arbeit wird bei zweiseitigen Barriere Optionen mit Rückvergütung der Fall payment at expiry vorausgesetzt.

Folgende Auszahlungen sind somit bei zweiseitigen Barriere Optionen am Ende der Laufzeit möglich:

	ohne Rückvergütung	zusätzliche Auszahlung bei Optionen mit Rückvergütung
knock-out	$h(A_N) \mathbb{1}_{\{l < \min_{n \in \{0, \dots, N\}} A_n, \max_{n \in \{0, \dots, N\}} A_n < r\}}$	$+R \mathbb{1}_{\{l \geq \min_{n \in \{0, \dots, N\}} A_n \vee \max_{n \in \{0, \dots, N\}} A_n \geq r\}}$
knock-in	$h(A_N) \mathbb{1}_{\{\min_{n \in \{0, \dots, N\}} A_n \leq l \vee \max_{n \in \{0, \dots, N\}} A_n \geq r\}}$	$+R \mathbb{1}_{\{l < \min_{n \in \{0, \dots, N\}} A_n, \max_{n \in \{0, \dots, N\}} A_n < r\}}$

Im Folgenden werden Barriere Optionen bewertet.

Für die Bewertung von knock-out Barriere Optionen ohne Rückvergütung benötigt man folgendes Lemma:

Lemma 3

Der eindimensionale Prozess $\{A_n\}_{n=0, \dots, N}$ erfüllt die Markov-Eigenschaft.

Beweis:

Sei $n \in \{0, \dots, N-1\}$ beliebig.

Sei $h : \mathbb{R} \rightarrow \mathbb{R}$ messbare Funktion, für die $E|h(A_{n+1})|$ existiert.

Es gilt: $h(A_{n+1}) = h(A_n \cdot Y_{n+1})$

Man definiere die Funktion $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, $f(Y_{n+1}, A_n) := h(A_n \cdot Y_{n+1}) = h(A_{n+1})$.

Dann gilt:

$$\begin{aligned}
 E[h(A_{n+1})|\mathcal{F}_n] &= E[f(Y_{n+1}, A_n)|\mathcal{F}_n] \\
 &\stackrel{\text{Lemma 0}}{=} E[f(Y_{n+1}, A_n)|A_n] \\
 &= E[h(A_{n+1})|A_n]
 \end{aligned}$$

Lemma 0 kann angewendet werden, da die Voraussetzungen des Lemmas erfüllt sind:

- $(\mathbb{R}, \mathfrak{B})$ ist messbarer Raum, (Ω, \mathcal{F}, P) ist Wahrscheinlichkeitsraum und \mathcal{F}_n Unter- σ -Algebra von \mathcal{F} .
- $Y_{n+1} : \Omega \rightarrow \mathbb{R}$, $A_n : \Omega \rightarrow \mathbb{R}$ sind messbare Abbildungen.
- $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, $f(Y_{n+1}, A_n) = h(A_{n+1})$ ist messbar.
- Y_{n+1} ist unabhängig von \mathcal{F}_n .
- A_n ist messbar bzgl. \mathcal{F}_n .
- $Ef(Y_{n+1}, A_n) = Eh(A_{n+1})$ existiert.

Somit erfüllt der eindimensionale Prozess $\{A_n\}_{n=0, \dots, N}$ die Markov-Eigenschaft. □

Bewertung von knock-out Barriere Optionen ohne Rückvergütung

Im Folgenden wird ein Algorithmus zur Bewertung von knock-out Barriere Optionen ohne Rückvergütung hergeleitet.

Sei $v_n(x)$ der Wert der Option in n , wenn $A_n = x$ gilt.

Sei P^* äquivalentes Martingalmaß mit $P^*(Y_i = u) = p^* = 1 - P^*(Y_i = d)$.

Man setze:

- $\tau := \inf\{n \in \mathbb{N}_0 | A_n \geq s\}$ für up-and-out Barriere Optionen
- $\tau := \inf\{n \in \mathbb{N}_0 | A_n \leq s\}$ für down-and-out Barriere Optionen
- $\tau := \inf\{n \in \mathbb{N}_0 | A_n \leq l \vee A_n \geq r\}$ für zweiseitige knock-out Barriere Optionen

Dann entspricht die Option dem Claim $C = \{C(n)\}_{n=1, \dots, N}$ mit $C(n) = 0$ für $n = 1, \dots, N - 1$ und $C(N) = h(A_N) \mathbb{1}_{\{\tau > N\}}$.

Es gilt:

$$\begin{aligned}
v_0(A_0) &= E^*[(1 + \rho)^{-N} \cdot C(N)|A_0] \\
&= (1 + \rho)^{-N} \cdot E^*[C(N)|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[C(N)|\mathcal{F}_n]|\mathcal{F}_0] \quad \text{für } 0 \leq n \leq N \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[h(A_N)\mathbb{1}_{\{\tau > N\}}|\mathcal{F}_n]|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[h(A_N)\mathbb{1}_{\{\tau > n\}}\mathbb{1}_{\{\tau > N\}}|\mathcal{F}_n]|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[\mathbb{1}_{\{\tau > n\}} \cdot E^*[h(A_N)\mathbb{1}_{\{\tau > N\}}|\mathcal{F}_n]|\mathcal{F}_0] \\
&\stackrel{\text{Lemma 3}}{=} (1 + \rho)^{-N} \cdot E^*[\mathbb{1}_{\{\tau > n\}} \cdot E^*[h(A_N)\mathbb{1}_{\{\tau > N\}}|A_n]|\mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[\mathbb{1}_{\{\tau > n\}} \cdot E^*[C(N)|A_n]|\mathcal{F}_0] \\
&= (1 + \rho)^{-n} \cdot E^*[\mathbb{1}_{\{\tau > n\}} \cdot E^*[(1 + \rho)^{-(N-n)} \cdot C(N)|A_n]|A_0] \\
&= (1 + \rho)^{-n} \cdot E^*[\mathbb{1}_{\{\tau > n\}} \cdot v_n(A_n)|A_0]
\end{aligned}$$

Für $n = 1$ folgt:

$$\begin{aligned}
v_0(A_0) &= (1 + \rho)^{-1} \cdot E^*[\mathbb{1}_{\{\tau > 1\}} \cdot v_1(A_1)|A_0] \\
&= \begin{cases} (1 + \rho)^{-1} \cdot (p^* \cdot v_1(uA_0) + (1 - p^*) \cdot v_1(dA_0)) & \text{falls } A_0 < s \text{ bzw. } A_0 > s \text{ bzw. } l < A_0 < r \\ 0 & \text{sonst} \end{cases}
\end{aligned}$$

Man erhält somit die Rekursionsformel:

$$\begin{aligned}
v_n(A_n) &= (1 + \rho)^{-1} \cdot E^*[\mathbb{1}_{\{\tau > n+1\}} \cdot v_{n+1}(A_{n+1})|A_n] \\
&= \begin{cases} (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(uA_n) + (1 - p^*) \cdot v_{n+1}(dA_n)) & \text{falls } A_n < s \text{ bzw. } A_n > s \text{ bzw. } l < A_n < r \\ 0 & \text{sonst} \end{cases}
\end{aligned}$$

für $n = N - 1, \dots, 0$.

Für den Wert der Option in N gilt: $v_N(A_N) = h(A_N)\mathbb{1}_{\{A_N < s\}}$ für up-and-out Barriere Optionen bzw. $v_N(A_N) = h(A_N)\mathbb{1}_{\{A_N > s\}}$ für down-and-out Barriere Optionen bzw. $v_N(A_N) = h(A_N)\mathbb{1}_{\{l < A_N < r\}}$ für zweiseitige knock-out Barriere Optionen. Man kann also für jeden möglichen Endkurs des Underlyings den Wert der Option in N berechnen.

Insgesamt ergibt sich der folgende Algorithmus zur Berechnung des Anfangspreises einer knock-out Barriere Option ohne Rückvergütung:

Initialisierung:

Für $i = 0, \dots, N$

berechne $v_N(A_0 \cdot u^i \cdot d^{N-i}) = v_N(A_N)$

$$= \begin{cases} h(A_N) \mathbb{1}_{\{A_N < s\}} & \text{für up-and-out Barriere Optionen} \\ h(A_N) \mathbb{1}_{\{A_N > s\}} & \text{für down-and-out Barriere Optionen} \\ h(A_N) \mathbb{1}_{\{l < A_N < r\}} & \text{für zweiseitige knock-out Barriere Optionen} \end{cases}$$

Rückwärtsinduktion:

Für $n = N - 1, \dots, 0$

Für $j = 0, \dots, n$

berechne $v_n(A_0 \cdot u^j \cdot d^{n-j}) = v_n(A_n)$

$$= \begin{cases} (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(uA_n) + (1 - p^*) \cdot v_{n+1}(dA_n)) & \text{falls } A_n < s \text{ bzw. } A_n > s \text{ bzw. } l < A_n < r \\ 0 & \text{sonst} \end{cases}$$

Return: $v_0(A_0)$

$v_0(A_0)$ ist der Anfangspreis der Option.

R-Programm (koBarriereOptionen.R)

Mit dem folgenden Programm kann der Anfangspreis einer knock-out Barriere Option ohne Rückvergütung berechnet werden. Dabei muss die Optionsart gewählt werden (up-and-out, down-and-out oder zweiseitige knock-out Barriere Option) und es müssen die Modell-Parameter A0, u, d, rho, N, die Auszahlungsfunktion h und die Schranken l und r bzw. s an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 1000$.

```
# Definition des CRR-Modells
A0<-1           # Anfangskurs des Underlyings
u<-1.2          # up-Parameter des CRR-Modells
d<-0.8          # down-Parameter des CRR-Modells
rho<-0.05       # Zinsrate pro Periode
N<-20           # Anzahl Perioden
```

```

p<-(1+rho-d)/(u-d) # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

# Definition der zu bewertenden Option
Optionsart <- "uo" # ko für zweiseitige knock-out Barriere Option,
                  # uo für up-and-out Barriere Option,
                  # do für down-and-out Barriere Option
l<-0.00001        # untere Schranke für zweiseitige knock-out Barriere Option
r<-20             # obere Schranke für zweiseitige knock-out Barriere Option
s<- 10           # Schranke für up-and-out bzw. down-and-out Barriere Option

# Definition der Auszahlungsfunktion
# bspw. Call
K<-1              # Basis für Call
h<-function(AN) {if(AN<K){return(0)}
               else {return(AN-K)}}
# AN Underlyingkurs in N

if (0<=p && p<=1) {
e <- numeric()
A <- matrix(e,N+1,N+1,F)
# A ist Matrix mit den möglichen Underlyingkursen
# A[i+1,n+1] enthält den Kurs nach n Perioden bei i ups

# Berechnen der möglichen Underlyingkurse für jede Periode
for (n in 0:N)
  for (i in 0:n) {A[i+1,n+1] <- A0*(u^i)*(d^(n-i))}

i1 <- seq(N-1,0,-1)

f <- numeric()
v <- matrix(f,N+1,N+1,F)
# v ist Matrix zur Berechnung der Werte der Option
# v[i+1,n+1] enthält den Wert der Option nach n Perioden bei i ups

# zweiseitige knock-out Barriere Option
# Initialisierung:
if (Optionsart == "ko")
  {for (i in 0:N)
    {if (A[i+1,N+1] <=1) {v[i+1,N+1] <- 0}
     else{if(A[i+1,N+1]>=r) {v[i+1,N+1] <- 0}
      else{v[i+1,N+1] <- h(A[i+1,N+1])}}}

# Rückwärtsinduktion:
for (n in i1)
  {for (i in 0:n)
    {if (A[i+1,n+1] <=1 ) {v[i+1,n+1] <- 0}
     else{if(A[i+1,n+1]>=r) {v[i+1,n+1] <- 0}

```

```

else{v[i+1,n+1] <- (1+rho)^(-1)*(p*v[i+2,n+2]+(1-p)*v[i+1,n+2])}}}}

# up-and-out Barriere Option
# Initialisierung:
if (Optionsart == "uo")
  {for (i in 0:N)
    {if (A[i+1,N+1]>=s) {v[i+1,N+1] <- 0}
     else{v[i+1,N+1] <- h(A[i+1,N+1])}}
# Rückwärtsinduktion:
  for (n in i1)
    {for (i in 0:n)
      {if (A[i+1,n+1]>=s) {v[i+1,n+1] <- 0}
       else{v[i+1,n+1] <- (1+rho)^(-1)*(p*v[i+2,n+2]+(1-p)*v[i+1,n+2])}}}}

# down-and-out Barriere Option
# Initialisierung:
if (Optionsart == "do")
  {for (i in 0:N)
    {if (A[i+1,N+1]<=s) {v[i+1,N+1] <- 0}
     else{v[i+1,N+1] <- h(A[i+1,N+1])}}
# Rückwärtsinduktion:
  for (n in i1)
    {for (i in 0:n)
      {if (A[i+1,n+1]<=s) {v[i+1,n+1] <- 0}
       else{v[i+1,n+1] <- (1+rho)^(-1)*(p*v[i+2,n+2]+(1-p)*v[i+1,n+2])}}}}

# Return:
cat ("Anfangswert ", Optionsart, v[1,1])}

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}

```

Bewertung von einseitigen knock-in Barriere Optionen ohne Rückvergütung

Sei $M_n := \max_{k \in \{0, \dots, n\}} A_k$ das laufende Maximum der Underlyingkurse und $m_n := \min_{k \in \{0, \dots, n\}} A_k$ das laufende Minimum der Underlyingkurse.

Dann entspricht eine up-and-in Barriere Option ohne Rückvergütung dem Claim

$C_1 = \{C_1(n)\}_{n=1, \dots, N}$ mit $C_1(n) = 0$ für $n = 1, \dots, N-1$ und $C_1(N) = h(A_N) \mathbb{1}_{\{M_N \geq s\}} =: g_1(A_N, M_N)$ und eine down-and-in Barriere Option ohne Rückvergütung dem Claim $C_2 = \{C_2(n)\}_{n=1, \dots, N}$ mit $C_2(n) = 0$ für $n = 1, \dots, N-1$ und $C_2(N) = h(A_N) \mathbb{1}_{\{m_N \leq s\}} =: g_2(A_N, m_N)$.

Es handelt sich somit bei einseitigen knock-in Barriere Optionen um spezielle Lookback Optionen. Die Bewertung erfolgt entsprechend.

R-Programme (up-and-in / down-and-in Barriere Option ohne Rückvergütung)

Der Anfangspreis einer up-and-in Barriere Option ohne Rückvergütung kann mit dem Programm LookbackMaxOptionen.R berechnet werden, indem man die Auszahlungsfunktion $g_1(A_N, M_N)$ verwendet.

Der Anfangspreis einer down-and-in Barriere Option ohne Rückvergütung kann mit dem Programm LookbackMinOptionen.R berechnet werden, indem man die Auszahlungsfunktion $g_2(A_N, m_N)$ verwendet.

Für die Bewertung von zweiseitigen knock-in Barriere Optionen benötigt man ein weiteres Lemma:

Lemma 4

Der dreidimensionale Prozess $\{(A_n, M_n, m_n)\}_{n=0, \dots, N}$ erfüllt die Markov-Eigenschaft.

Beweis:

Sei $n \in \{0, \dots, N-1\}$ beliebig.

Sei $h : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ messbare Funktion, für die $E|h(A_{n+1}, M_{n+1}, m_{n+1})|$ existiert.

$$\begin{aligned} \text{Es gilt: } h(A_{n+1}, M_{n+1}, m_{n+1}) &= h(A_{n+1}, M_n \vee A_{n+1}, m_n \wedge A_{n+1}) \\ &= h(A_n \cdot Y_{n+1}, M_n \vee (A_n \cdot Y_{n+1}), m_n \wedge (A_n \cdot Y_{n+1})) \end{aligned}$$

Man definiere die Funktion $f : \mathbb{R} \times (\mathbb{R} \times \mathbb{R} \times \mathbb{R}) \rightarrow \mathbb{R}$,

$$f(Y_{n+1}, (A_n, M_n, m_n)) := h(A_n \cdot Y_{n+1}, M_n \vee (A_n \cdot Y_{n+1}), m_n \wedge (A_n \cdot Y_{n+1})) = h(A_{n+1}, M_{n+1}, m_{n+1}).$$

Dann gilt:

$$\begin{aligned} E[h(A_{n+1}, M_{n+1}, m_{n+1})|\mathcal{F}_n] &= E[f(Y_{n+1}, (A_n, M_n, m_n))|\mathcal{F}_n] \\ &\stackrel{\text{Lemma 0}}{=} E[f(Y_{n+1}, (A_n, M_n, m_n))|(A_n, M_n, m_n)] \\ &= E[h(A_{n+1}, M_{n+1}, m_{n+1})|(A_n, M_n, m_n)] \end{aligned}$$

Lemma 0 kann angewendet werden, da die Voraussetzungen des Lemmas erfüllt sind:

- $(\mathbb{R}, \mathfrak{B})$, $(\mathbb{R} \times \mathbb{R} \times \mathbb{R}, \mathfrak{B} \otimes \mathfrak{B} \otimes \mathfrak{B})$ sind messbare Räume, (Ω, \mathcal{F}, P) ist Wahrscheinlichkeitsraum und \mathcal{F}_n Unter- σ -Algebra von \mathcal{F} .
- $Y_{n+1} : \Omega \rightarrow \mathbb{R}$, $(A_n, M_n, m_n) : \Omega \rightarrow \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ sind messbare Abbildungen.

- $f : \mathbb{R} \times (\mathbb{R} \times \mathbb{R} \times \mathbb{R}) \rightarrow \mathbb{R}$,
 $f(Y_{n+1}, (A_n, M_n, m_n)) = h(A_{n+1}, M_{n+1}, m_{n+1})$ ist messbar.
- Y_{n+1} ist unabhängig von \mathcal{F}_n .
- (A_n, M_n, m_n) ist messbar bzgl. \mathcal{F}_n .
- $Ef(Y_{n+1}, (A_n, M_n, m_n)) = Eh(A_{n+1}, M_{n+1}, m_{n+1})$ existiert.

Somit erfüllt der dreidimensionale Prozess $\{(A_n, M_n, m_n)\}_{n=0, \dots, N}$ die Markov-Eigenschaft. □

Bewertung von zweiseitigen knock-in Barriere Optionen ohne Rückvergütung

Im Folgenden wird ein Algorithmus zur Bewertung von zweiseitigen knock-in Barriere Optionen ohne Rückvergütung hergeleitet.

Sei $v_n(x, y, z)$ der Wert der Option in n , wenn $A_n = x$, $M_n = y$, $m_n = z$ gilt.

Sei P^* äquivalentes Martingalmaß mit $P^*(Y_i = u) = p^* = 1 - P^*(Y_i = d)$.

Eine zweiseitige knock-in Barriere Option ohne Rückvergütung entspricht dem Claim

$$C = \{C(n)\}_{n=1, \dots, N} \text{ mit } C(n) = 0 \text{ für } n = 1, \dots, N-1 \text{ und } C(N) = h(A_N) \mathbb{1}_{\{m_N \leq l \vee M_N \geq r\}} =: g(A_N, M_N, m_N).$$

Es gilt:

$$\begin{aligned}
v_0(A_0, M_0, m_0) &= E^*[(1 + \rho)^{-N} \cdot C(N) | (A_0, M_0, m_0)] \\
&= (1 + \rho)^{-N} \cdot E^*[C(N) | \mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[C(N) | \mathcal{F}_1] | \mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[g(A_N, M_N, m_N) | \mathcal{F}_1] | \mathcal{F}_0] \\
&\stackrel{\text{Lemma 4}}{=} (1 + \rho)^{-N} \cdot E^*[E^*[g(A_N, M_N, m_N) | (A_1, M_1, m_1)] | \mathcal{F}_0] \\
&= (1 + \rho)^{-N} \cdot E^*[E^*[C(N) | (A_1, M_1, m_1)] | \mathcal{F}_0] \\
&= (1 + \rho)^{-1} \cdot E^*[E^*[(1 + \rho)^{-(N-1)} \cdot C(N) | (A_1, M_1, m_1)] | (A_0, M_0, m_0)] \\
&= (1 + \rho)^{-1} \cdot E^*[v_1(A_1, M_1, m_1) | (A_0, M_0, m_0)] \\
&= (1 + \rho)^{-1} \cdot (p^* \cdot v_1(u \cdot A_0, M_0 \vee u \cdot A_0, m_0 \wedge u \cdot A_0) \\
&\quad + (1 - p^*) \cdot v_1(d \cdot A_0, M_0 \vee d \cdot A_0, m_0 \wedge d \cdot A_0)) \\
&= (1 + \rho)^{-1} \cdot (p^* \cdot v_1(u \cdot A_0, M_0 \vee u \cdot A_0, m_0) \\
&\quad + (1 - p^*) \cdot v_1(d \cdot A_0, M_0, m_0 \wedge d \cdot A_0))
\end{aligned}$$

Somit erhält man die Rekursionsformel:

$$v_n(A_n, M_n, m_n) = (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, M_n \vee u \cdot A_n, m_n) + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, M_n, m_n \wedge d \cdot A_n)) \text{ für } n = N - 1, \dots, 0.$$

Der Wert einer zweiseitigen knock-in Barriere Option in N entspricht der Auszahlung der Option in N, d.h. $v_N(A_N, M_N, m_N) = g(A_N, M_N, m_N)$.

Insgesamt ergibt sich der folgende Algorithmus zur Berechnung des Anfangspreises einer zweiseitigen knock-in Barriere Option ohne Rückvergütung:

Initialisierung:

Für jedes mögliche Tripel (A_N, M_N, m_N)

berechne $v_N(A_N, M_N, m_N) = g(A_N, M_N, m_N) = h(A_N) \mathbb{1}_{\{m_N \leq l \vee M_N \geq r\}}$

Rückwärtsinduktion:

Für $n = N - 1, \dots, 0$

Für jedes mögliche Tripel (A_n, M_n, m_n)

berechne $v_n(A_n, M_n, m_n) = (1 + \rho)^{-1} \cdot (p^* \cdot v_{n+1}(u \cdot A_n, M_n \vee u \cdot A_n, m_n) + (1 - p^*) \cdot v_{n+1}(d \cdot A_n, M_n, m_n \wedge d \cdot A_n))$

Return: $v_0(A_0, M_0, m_0)$

$v_0(A_0, M_0, m_0)$ ist der Anfangspreis der Option.

R-Programm (zskiBarriereOptionen.R)

Mit dem folgenden Programm kann der Anfangspreis einer zweiseitigen knock-in Barriere Option ohne Rückvergütung berechnet werden. Dabei müssen die Modell-Parameter A0, u, d, rho, N, die Auszahlungsfunktion h und die Barrieren l und r an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 20$.

```
# Definition des CRR-Modells
A0<-1           # Anfangskurs des Underlyings
u<-1.5          # up-Parameter des CRR-Modells
d<-0.6          # down-Parameter des CRR-Modells
rho<-0.05       # Zinsrate pro Periode
N<-15           # Anzahl Perioden
```



```

p<-(1+rho-d)/(u-d) # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

l<- 0.5           # untere Schranke
r<- 2            # obere Schranke
R<-0            # (Rückvergütung)

# Definition Auszahlungsfunktion
# bspw. Call
K<-1            # Basis für Call
h<-function(A) {if(A<K){return(0)}
else {return(A-K)}}
# A Underlyingkurs in N

if (0<=p && p<=1) {
e <- numeric()
Aus <- array(e, dim=c(2^(N+1)-1,5))
# Spalten: Periodenanzahl, Underlyingkurs, Maximum Underlyingkurse,
#           Minimum Underlyingkurse, Optionswert
# Speicherung in Aus: Die Werte von Periode n stehen von Zeile 2^n bis 2^(n+1)-1
# Steht An in Zeile 2^n+i, so steht dAn in Zeile 2^(n+1)+2i+1
#                       und uAn in Zeile 2^(n+1)+2i

# Speichern Anfangskurs
Aus[1,1] <- 0
Aus[1,2] <- A0
Aus[1,3] <- A0
Aus[1,4] <- A0

# Ermittlung der möglichen Auszahlungen, Maxima und Minima
Zeile <- 1      # Zeile im Array Aus

for (n in 1:N)
  {for (i in (2^(n-1)):(2^n-1))
    {Zeile <- Zeile + 1
     Aus[Zeile,1] <- n
     Aus[Zeile,2] <- u * Aus[i,2]
     Aus[Zeile,3] <- max(Aus[i,3],Aus[Zeile,2])
     Aus[Zeile,4] <- min(Aus[i,4],Aus[Zeile,2])

     Zeile <- Zeile + 1
     Aus[Zeile,1] <- n
     Aus[Zeile,2] <- d * Aus[i,2]
     Aus[Zeile,3] <- max(Aus[i,3],Aus[Zeile,2])
     Aus[Zeile,4] <- min(Aus[i,4],Aus[Zeile,2]) }}

# Initialisierung:

```

```

for (i in (2^(N)):(2^(N+1)-1))
  {if (Aus[i,3] >= r | Aus[i,4] <= 1) {Aus[i,5] <- h(Aus[i,2])}
   else {Aus[i,5] <- R}}

# Rückwärtsinduktion:
for (n in seq(N-1,0,-1))
  {for (i in (2^(n)):(2^(n+1)-1))
   {Zeilevu <- 2^(n+1) + 2 * ( i - 2^n )
    Zeilevd <- Zeilevu + 1
    Aus[i,5] <- (1+rho)^(-1)*(p*Aus[Zeilevu,5]+(1-p)*Aus[Zeilevd,5]) }}

# Return:
cat ("Anfangspreis", Aus[1,5])}

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}

```

Bewertung von knock-in Barriere Optionen mit Rückvergütung

Eine up-and-in Barriere Option mit Rückvergütung entspricht dem Claim $C_1 = \{C_1(n)\}_{n=1,\dots,N}$ mit

$C_1(n) = 0$ für $n = 1, \dots, N-1$ und $C_1(N) = h(A_N)\mathbb{1}_{\{M_N \geq s\}} + R\mathbb{1}_{\{M_N < s\}} =: g_1(A_N, M_N)$,

eine down-and-in Barriere Option mit Rückvergütung dem Claim $C_2 = \{C_2(n)\}_{n=1,\dots,N}$ mit

$C_2(n) = 0$ für $n = 1, \dots, N-1$ und $C_2(N) = h(A_N)\mathbb{1}_{\{m_N \leq s\}} + R\mathbb{1}_{\{m_N > s\}} =: g_2(A_N, m_N)$ und

eine zweiseitige knock-in Barriere Option mit Rückvergütung dem Claim $C_3 = \{C_3(n)\}_{n=1,\dots,N}$ mit $C_3(n) = 0$ für $n = 1, \dots, N-1$ und $C_3(N) = h(A_N)\mathbb{1}_{\{m_N \leq l \vee M_N \geq r\}} + R\mathbb{1}_{\{m_N > l, M_N < r\}} =: g_3(A_N, M_N, m_N)$.

Es handelt sich somit bei einseitigen knock-in Barriere Optionen mit Rückvergütung um spezielle Lookback Optionen mit entsprechender Bewertung. Zweiseitige knock-in Barriere Optionen mit Rückvergütung können mit dem gleichen Algorithmus wie zweiseitige knock-in Barriere Optionen ohne Rückvergütung bewertet werden.

R-Programme (knock-in Barriere Optionen mit Rückvergütung)

Der Anfangspreis einer up-and-in Barriere Option mit Rückvergütung kann mit dem Programm LookbackMaxOptionen.R berechnet werden, indem man die Auszahlungsfunktion $g_1(A_N, M_N)$ verwendet,

der Anfangspreis einer down-and-in Barriere Option mit Rückvergütung kann mit dem Programm LookbackMinOptionen.R berechnet werden, indem man die Auszahlungsfunktion $g_2(A_N, m_N)$ verwendet und

der Anfangspreis einer zweiseitigen knock-in Barriere Option mit Rückvergütung kann mit dem Programm zskiBarriereOptionen.R berechnet werden, indem man die Auszahlungsfunktion $g_3(A_N, M_N, m_N)$ verwendet und den Parameter R anpasst.

Bewertung von einseitigen knock-out Barriere Optionen mit Rückvergütung

Eine up-and-out Barriere Option mit Rückvergütung entspricht dem Claim $C_1 = \{C_1(n)\}_{n=1,\dots,N}$ mit

$$C_1(n) = 0 \text{ für } n = 1, \dots, N-1 \text{ und } C_1(N) = h(A_N)\mathbb{1}_{\{M_N < s\}} + R\mathbb{1}_{\{M_N \geq s\}} =: g_1(A_N, M_N)$$

und eine down-and-out Barriere Option mit Rückvergütung entspricht dem Claim

$$C_2 = \{C_2(n)\}_{n=1,\dots,N} \text{ mit}$$

$$C_2(n) = 0 \text{ für } n = 1, \dots, N-1 \text{ und } C_2(N) = h(A_N)\mathbb{1}_{\{m_N > s\}} + R\mathbb{1}_{\{m_N \leq s\}} =: g_2(A_N, M_N).$$

Es handelt sich somit bei einseitigen knock-out Barriere Optionen mit Rückvergütung um spezielle Lookback Optionen. Die Bewertung erfolgt entsprechend.

R-Programme (up-and-out / down-and-out Barriere Option mit Rückvergütung)

Der Anfangspreis einer up-and-out Barriere Option mit Rückvergütung kann mit dem Programm LookbackMaxOptionen.R berechnet werden, indem man die Auszahlungsfunktion $g_1(A_N, M_N)$ verwendet und

der Anfangspreis einer down-and-out Barriere Option mit Rückvergütung kann mit dem Programm LookbackMinOptionen.R berechnet werden, indem man die Auszahlungsfunktion $g_2(A_N, m_N)$ verwendet.

Bewertung von zweiseitigen knock-out Barriere Optionen mit Rückvergütung

Eine zweiseitige knock-out Barriere Option mit Rückvergütung entspricht dem Claim

$$C_3 = \{C_3(n)\}_{n=1,\dots,N}$$

$$\text{mit } C_3(n) = 0 \text{ für } n = 1, \dots, N-1 \text{ und } C_3(N) = h(A_N)\mathbb{1}_{\{l < m_N, M_N < r\}} + R\mathbb{1}_{\{m_N \leq l \vee M_N \geq r\}} =: g_3(A_N, M_N, m_N).$$

Zweiseitige knock-out Barriere Optionen mit Rückvergütung können somit mit dem Algorithmus bewertet werden, der für zweiseitige knock-in Barriere Optionen hergeleitet wurde.

R-Programm (zskoBarriereOptionen.R)

Mit dem folgenden Programm kann der Anfangspreis einer zweiseitigen knock-out Barriere Option mit Rückvergütung berechnet werden. Dabei müssen die Modell-Parameter A_0 , u , d , ρ , N , die Auszahlungsfunktion h , die Schranken l und r und die Rückvergütung R an die Option angepasst werden. Das Programm läuft bis zu einer Periodenanzahl von ungefähr $N = 20$.

```
# Definition des CRR-Modells
A0<-1          # Anfangskurs des Underlyings
u<-1.5         # up-Parameter des CRR-Modells
d<-0.6         # down-Parameter des CRR-Modells
rho<-0.05      # Zinsrate pro Periode
N<-15         # Anzahl Perioden
p<-(1+rho-d)/(u-d) # Wahrscheinlichkeit für up (äquivalentes Martingalmaß)

l<- 0.5        # untere Schranke
r<- 2         # obere Schranke
R<- 1         # Rückvergütung

# Definition Auszahlungsfunktion
# bspw. Call
K<-1          # Basis für Call
h<-function(A) {if(A<K){return(0)}
else {return(A-K)}}
# A Underlyingkurs in N

if (0<=p && p<=1) {
e <- numeric()
Aus <- array(e, dim=c(2^(N+1)-1,5))
# Spalten: Periodenanzahl, Underlyingkurs, Maximum Underlyingkurse,
#          Minimum Underlyingkurse, Optionswert
# Speicherung in Aus: Die Werte von Periode n stehen von Zeile 2^n bis 2^(n+1)-1
# Steht An in Zeile 2^n+i, so steht dAn in Zeile 2^(n+1)+2i+1
#                      und uAn in Zeile 2^(n+1)+2i

# Speichern Anfangskurs
Aus[1,1] <- 0
Aus[1,2] <- A0
Aus[1,3] <- A0
Aus[1,4] <- A0

# Ermittlung der möglichen Auszahlungen, Maxima und Minima
Zeile <- 1      # Zeile im Array Aus

for (n in 1:N)
```

```

{for (i in (2^(n-1)): (2^n-1))
  {Zeile <- Zeile + 1
   Aus[Zeile,1] <- n
   Aus[Zeile,2] <- u * Aus[i,2]
   Aus[Zeile,3] <- max(Aus[i,3], Aus[Zeile,2])
   Aus[Zeile,4] <- min(Aus[i,4], Aus[Zeile,2])

   Zeile <- Zeile + 1
   Aus[Zeile,1] <- n
   Aus[Zeile,2] <- d * Aus[i,2]
   Aus[Zeile,3] <- max(Aus[i,3], Aus[Zeile,2])
   Aus[Zeile,4] <- min(Aus[i,4], Aus[Zeile,2]) }}

# Initialisierung:
for (i in (2^N):(2^(N+1)-1))
  {if (1 < Aus[i,4] && Aus[i,3] < r) {Aus[i,5] <- h(Aus[i,2])}
   else {Aus[i,5] <- R}}

# Rückwärtsinduktion:
for (n in seq(N-1,0,-1))
  {for (i in (2^n):(2^(n+1)-1))
   {Zeilevu <- 2^(n+1) + 2 * ( i - 2^n )
    Zeilevd <- Zeilevu + 1
    Aus[i,5] <- (1+rho)^(-1)*(p*Aus[Zeilevu,5]+(1-p)*Aus[Zeilevd,5]) }}

# Return:
cat ("Anfangspreis", Aus[1,5])}

if (p<0 | p >1) {cat("unzulässige Modell-Parameter u, d, rho")}

```

Fazit

Wie diese Arbeit zeigt, können die betrachteten pfadabhängigen exotischen Optionen im CRR Modell mithilfe der Markov-Eigenschaft geeigneter stochastischer Prozesse bewertet werden. Dabei werden die Anfangspreise der Optionen durch Initialisierung und Rückwärtsinduktion berechnet. Dafür werden für jede Periode und jeden möglichen Underlyingkurs zusätzliche optionsartabhängige Informationen über den bisherigen Kursverlauf benötigt, z.B. mögliche Summen der Underlyingkurse für asiatische Optionen oder mögliche Maxima der Underlyingkurse für LookbackMaxOptionen.

Bei der Implementierung der Algorithmen zur Optionsbewertung in R besteht die Schwierigkeit darin, diese zusätzlichen Informationen mit möglichst wenig Speicher und möglichst geringer Laufzeit zu erhalten und bei der Rückwärtsinduktion in den berechneten Werten jeweils die zwei benötigten Werte möglichst effizient zu finden. Dadurch haben die einzelnen Programme sehr unterschiedliche maximale Periodenanzahlen von ungefähr 20, z.B. für asiatische Optionen und zweiseitige knock-in Barriere Optionen, bis ungefähr 1000 für knock-out Barriere Optionen ohne Rückvergütung.

Literaturverzeichnis

- [1] Steven Shreve: *Stochastic Calculus and Finance*, 1996.
- [2] Klaus Sandmann: *Einführung in die Stochastik der Finanzmärkte*, Springer-Verlag, 2010.
- [3] Albrecht Irle: *Finanzmathematik Die Bewertung von Derivaten*, Teubner-Verlag, 2003.
- [4] http://en.wikipedia.org/wiki/Russian_option,
01.08.2010

Gemäß § 11 Absatz 5 der Prüfungsordnung für den Bachelorstudiengang Mathematik an der Westfälischen Wilhelms-Universität versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Münster, den 11.08.2010